(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 056 010 A1

(12) # EUROPEAN PATENT APPLICATION

(43) Date of publication:
29.11.2000 Bulletin 2000/48

(51) Int Cl.7: **G06F 11/00**, G06F 1/00,
G06F 12/14

(21) Application number: 99304166.4

(22) Date of filing: 28.05.1999

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(71) Applicant: Hewlett-Packard Company
Palo Alto, California 94304-1112 (US)

(72) Inventors:
- Proudler, Graeme John
Bristol BS34 8XQ (GB)
- Balacheff, Boris
Bristol BS31 2HJ (GB)

(74) Representative: Lawman, Matthew John Mitchell
Hewlett-Packard Limited,
IP Section,
Building 3,
Filton Road
Stoke Gifford, Bristol BS34 8QZ (GB)

(54) **Data integrity monitoring in trusted computing entity**

(57) A method of security monitoring of data files in a computer platform is carried out by a trusted component having a processor and trusted memory area. The method comprises creating one or a plurality of data files in an untrusted memory area of said computing platform, for each created data file, periodically generating a digest data by applying a hash function to each data file, storing the digest data in a trusted memory area and for each file periodically comparing a current digest data of the file with a previously generated digest data of the file. Any differences between a previous and a current digest data indicate that a file in the untrusted memory area has been corrupted.

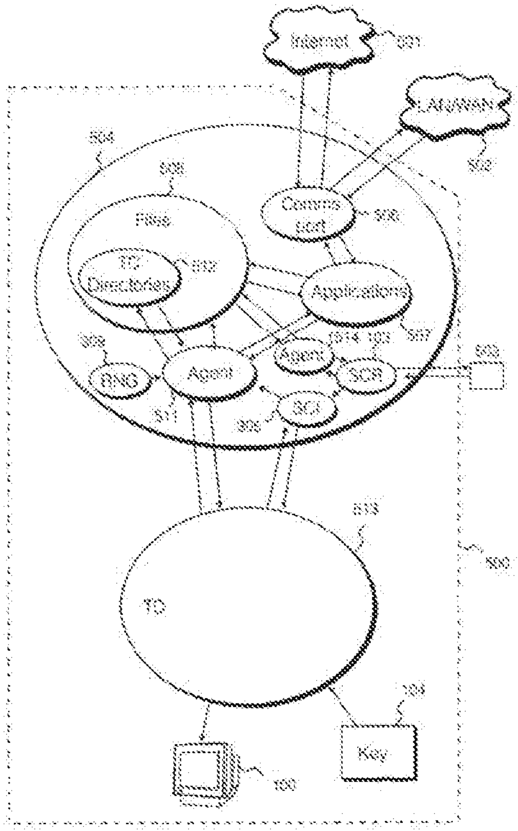


Fig. 5

Printed by Jouve, 75001 PARIS (FR)

Description

Field of the Invention

[0001] The present invention relates to a computing platform, and particularly, although not exclusively, to methods and apparatus for verifying a state of data integrity of the computing platform.

Background to the Invention

[0002] Conventional prior art mass market computing platforms include the well-known personal computer (PC) and competing products such as the Apple Macintosh™, and a proliferation of known palm-top and laptop personal computers. Generally, markets for such machines fall into two categories, these being domestic or consumer, and corporate. A general requirement for a computing platform for domestic or consumer use is a relatively high processing power, Internet access features, and multi-media features for handling computer games. For this type of computing platform, the Microsoft Windows® '95 and '98 operating system products and Intel processors dominate the market.

[0003] On the other hand, for business use, there are a plethora of available proprietary computer platform solutions available aimed at organizations ranging from small businesses to multi-national organizations. In many of these applications, a server platform provides centralized data storage, and application functionality for a plurality of client stations. For business use, other key criteria are reliability, networking features, and security features. For such platforms, the Microsoft Windows NT 4.0™ operating system is common, as well as the Unix™ operating system.

[0004] With the increase in commercial activity transacted over the Internet, known as "e-commerce", there has been much interest in the prior art on enabling data transactions between computing platforms, over the Internet. However, because of the potential for fraud and manipulation of electronic data, in such proposals, fully automated transactions with distant unknown parties on a wide-spread scale as required for a fully transparent and efficient market place have so far been held back. The fundamental issue is one of trust between interacting computer platforms for the making of such transactions.

[0005] There have been several prior art schemes which are aimed at increasing the security and trustworthiness of computer platforms. Predominantly, these rely upon adding in security features at the application level, that is to say the security features are not inherently imbedded in the kernel of operating systems, and are not built in to the fundamental hardware components of the computing platform. Portable computer devices have already appeared on the market which include a smart card, which contains data specific to a user, which is input into a smart card reader on the computer. Presently, such smart cards are at the level of being add-on extras to conventional personal computers, and in some cases are integrated into a casing of a known computer. Although these prior art schemes go some way to improving the security of computer platforms, the levels of security and trustworthiness gained by prior art schemes may be considered insufficient to enable widespread application of automated transactions between computer platforms. Before businesses expose significant value transactions to electronic commerce on a widespread scale, they may require greater confidence in the trustworthiness of the underlying technology.

[0006] In the applicant's co-pending disclosures 'Trusted Computing Platform', filed at the European Patent Office on 15 February 1999, the entire contents of which are incorporated herein by reference, and 'Computing Apparatus and Methods of Operating Computing Apparatus', there is disclosed a concept of a 'trusted computing platform' comprising a computing platform which has a 'trusted component' in the form of a built-in hardware and software component. Two computing entities each provisioned with such a trusted component, may interact with each other with a high degree of 'trust'. That is to say, where the first and second computing entities interact with each other the security of the interaction is enhanced compared to the case where no trusted component is present, because

- A user of a computing entity has higher confidence in the integrity and security of his/her own computer entity and in the integrity and security of the computer entity belonging to the other computing entity.

- Each entity is confident that the other entity is in fact the entity which it purports to be.

- Where one or both of the entities represent a party to a transaction, e.g. a data transfer transaction, because of the in-built trusted component, third party entities interacting with the entity have a high degree of confidence that the entity does in fact represent such a party.

- The trusted component increases the inherent security of the entity itself, through verification and monitoring processes implemented by the trusted component.

- The computer entity is more likely to behave in the way it is expected to behave.

[0007] Prior art computing platforms have several problems which need to be overcome in order to realize the potential of the applicants' above disclosed trusted component concept. In particular,

- The operating status of a computer system or platform and the status of the data within the platform or system is dynamic and difficult to predict. It is difficult to determine whether a computer platform is operating correctly because the state of the computer platform and data on the platform is constantly changing and the computer platform itself may be dynamically changing.

- From a security point of view, commercial computer platforms, in particular client platforms, are often deployed in environments which are vulnerable to unauthorized modification. The main areas of vulnerability include modification by software loaded by a user, or by software loaded via a network connection. Particularly, but not exclusively, conventional computer platforms may be vulnerable to attack by virus programs, with varying degrees of hostility.

- Computer platforms may be upgraded or their capabilities extended or restricted by physical modification, i.e. addition or deletion of components such as hard disk drives, peripheral drivers and the like.

[0008] In particular, conventional computer platforms are susceptible to attack by computer viruses, of which there are thousands of different varieties. Several proprietary virus finding and correcting applications are known, for example the Dr Solomons ™ virus toolkit program, and the Microsoft ™ virus guard facility provided within the Windows ™ operating system. However, such virus packages protect primarily against known viruses, and new strains of virus are being developed and released into the computing and internet environment on an ongoing basis.

Summary of the Invention

[0009] In one specific form, the invention provides a computer platform with a trusted component which generates integrity metrics describing the integrity of data on the computer platform, which can be reported to a user of the computer platform, or to a third party entity communicating with the computer platform, for example over a network connection.

[0010] Suitably the integrity metrics are dynamic metrics, which can provide continuous, or regular monitoring of the computer platform during its operation.

[0011] Methods for measuring and reporting the dynamic integrity metrics are operated partly within a trusted component, and partly within a computer platform being monitored by the trusted component.

[0012] According to first aspect of the present invention is provided a method of security monitoring of a computer platform, said method comprising the steps of:

(i) creating a data file in a memory area of said computing platform;
(ii) generating a first digest data describing a data content of said data file;
(iii) waiting a predetermined time period;
(iv) repeating step (ii) to generate a second digest data; and
(v) comparing said second digest data with said first digest data.

[0013] Preferably if second digest data is identical to said first digest data said steps i) to v) above are repeated.

[0014] If said second digest data is not identical to said first digest data, an error data is stored in said trusted memory area.

[0015] Preferably said step of generating a first digest data comprises applying a hash function to said data file to produce a hash function data corresponding to said data file.

[0016] Said step of creating a data file in a memory area of said computer platform may comprise copying an existing user data file into a reserved portion of said memory area of said computer platform.

[0017] Said step of creating a data file in said memory area may comprise generating a random or pseudo random data in a reserved portion of said memory area of said computer platform.

[0018] Preferably step of generating a digest data corresponding to said data file is carried out by an algorithm operating on said computer platform.

[0019] Said step of generating a digest data may comprise sending a said data file to a trusted component comprising a trusted processor and a trusted memory area, and generating said digest data by applying an algorithm to said data file in said trusted component.

[0020] According to a second aspect of the present invention there is provided a computer entity comprising:

a computer platform comprising a first data processing means and a first memory means;

a monitoring component comprising a second data processing means and a second memory means;

wherein said monitoring component comprises means for receiving a monitor data, said monitor data describing a content of a plurality of data files stored in said computer platform in said first memory means;

means for storing said plurality of monitor data in said monitoring component; and

means for making comparisons of said monitor data,

wherein said monitoring component receives for each of a plurality of data files, an historical monitor data representing a state of said data file at a previous point in time, and a current monitor data representing a current state of said data file.

[0021] Preferably said historical monitor data and said current monitor data are stored in said second memory means of said monitoring component.

[0022] Preferably said monitoring component comprises a set of agent code stored in said second data storage means, wherein said set of agent code may be transferred to said first data storage means for operation and control by said first data processing means in said computer platform.

[0023] Preferably said monitoring component comprises a dictionary means, said dictionary means comprising a text generator device operable to generate a plurality of text data representing a plurality of words, and said monitoring means transferring said text data to a plurality of said data files created in a reserved area of said first memory means.

[0024] Preferably said dictionary means is operable to generate a plurality of names identifying said plurality of data files created in said reserved area of said first memory means.

[0025] Preferably said dictionary means is operable to generate a plurality of names of directories containing said plurality of data files created in said reserved area of said first memory means.

[0026] Preferably the computer entity further comprises an agent means, said agent means resident and operating on said computer platform wherein,

said agent means creates a plurality of said data files in a reserved region of said first memory area;

said agent means substantially continuously monitors said created data files in said reserved memory region; and

said agent reports said monitor data describing a content of said data files in said reserved memory region periodically to said monitoring component.

[0027] Said computer entity may comprise a random data generator, wherein said random data generator generates random data which is stored in a plurality of said data files created in a reserved region of said first memory area of said computer platform.

[0028] Said computer entity may comprise an agent device resident on said computer platform, and a smart card reader device, wherein said agent device communicates with said smart card reader device for receiving a file name data from a smartcard via said smart card reader device, said file name data describing one or a plurality of file names of user data files for which permission to copy said user data files is granted to said agent device.

[0029] According to third aspect of the present invention there is provided a method of security monitoring a computer platform comprising a first data processing means and a first memory means, said method comprising the steps of:

receiving a first monitor data, said first monitor data describing a data content of a data file stored in said computer platform;

storing said first monitor data in a trusted memory area physically and logically distinct from said computer platform;

receiving a second monitor data, said second monitor data describing a data content of said same data file stored in said computer platform;

comparing said first monitor data with said second monitor data; and

if said first monitor data differs from said second monitor data, generating an error data.

[0030] Preferably said method further comprises the step of generating said monitor data by applying a one-way function algorithm to a data content of said data file.

[0031] Preferably an alarm display is generated when a said error data is created.

[0032] The method may comprise the step of:

comparing said error data against a predetermined measure of error data allowable in a predetermined time, to determine if said error data is statistically significant.

[0033] If said error data is determined to be statistically significant, an alarm display may be generated indicating an error has occurred in said data file.

[0034] The invention includes a method of monitoring a computer platform comprising a first data processing means and first memory means, said method comprising the steps of:

a) allocating a region of said first memory means for use by a monitoring entity comprising a second data processing means and a second memory means;

b) creating in said allocated memory area a plurality of data files, each allocated to said monitoring entity;

c) entering data into said plurality of allocated data files in said reserved memory region;

d) creating for each of said data files a monitor data describing a data content of each of said data file;

e) storing said monitor data in a second memory device, said second memory device being physically and logically distinct from said first memory device;

f) repeating steps d) and e); and

g) periodically comparing a recently received said monitor data for said data file with a previously received monitor data for the same said data file.

[0035] Said step of entering data into a said data file may comprise:

copying an existing data file from an unreserved area of said first memory device into said data file.

[0036] Said method may further comprise the step of:

assigning a file name of said data file in said reserved memory area said file name being a different file name to a file name of said original user file from said unreserved area of said first memory area from which said data file was copied.

[0037] Said method may further comprise the step of:

assigning a directory name of a directory used for storing said data file in said reserved memory area said directory name being a different directory name to a directory name of said original user directory from said unreserved area of said first memory area in which said data file was originally located.

[0038] Preferably said step of creating a monitor data comprises:

applying a one-way function algorithm to data in said data file, to produce said monitor data from said data stored in said data file.

Brief Description of the Drawings

[0039] For a better understanding of the invention and to show how the same may be carried into effect, there will now be described by way of example only, specific embodiments, methods and processes according to the present invention with reference to the accompanying drawings in which:

Fig. 1 illustrates schematically a computer entity according to first specific embodiment of the present invention;

Fig. 2 illustrates schematically connectivity of selected components of the computer entity of Fig. 1;

Fig. 3 illustrates schematically a hardware architecture of components of the computer entity of Fig. 1;

Fig. 4 illustrates schematically an architecture of a trusted component comprising the computer entity of Fig. 1;

Fig. 5 illustrates schematically a logical architecture of the computer entity of Fig. 1, comprising a trusted space, and a user space;

Fig 6 illustrates schematically a logical content of a software agent component of the computer entity of Fig. 1;

Fig. 7 illustrates schematically a set of logical components comprising a trusted component of the computer entity of Fig. 1;

Fig. 8 illustrates schematically a dialog box display generated by the trusted component;

Fig. 9 illustrates a further dialog display generated by the trusted component;

Fig. 10 illustrates schematically a first set of process steps carried out by the trusted component and a software agent of the trusted component for performing a security monitoring method according to the present invention;

Fig. 11 illustrates schematically a second set of process steps carried out by the trusted component and software agent for carrying out a security monitoring method according to the present invention; and

Fig. 12 illustrates schematically a third set of process steps and interaction between the trusted component, its software agent, a user smart card, and its software agent for carrying out a specific method according to the present invention.

Detailed Description of the Best Mode for Carrying Out the Invention

[0040] There will now be described by way of example the best mode contemplated by the inventors for carrying out the invention. In the following description numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent however, to one skilled in the art, that the present invention may be practiced without limitation to these specific details. In other instances, well known methods and structures have not been described in detail so as not to unnecessarily obscure the present invention.

[0041] Referring to Fig. 1 herein, there is illustrated schematically one example of a trusted computer entity as previously described in the applicant's European patent application entitled "Trusted Computing Platform", filed 15 February 1999 at the European Patent Office a copy of which is filed herewith, and the entire contents of which are incorporated herein by reference. Referring to Fig. 2 of the accompanying drawings, there is illustrated schematically physical connectivity of some of the components of the trusted computer entity of Fig. 1. Referring to Fig. 3 herein, there is illustrated schematically an architecture of the trusted computer entity of Figs. 1

and 2, showing physical connectivity of components of the entity.

[0042] In general, in the best mode described herein, a trusted computing entity comprises a computer platform consisting of a first data processor, and a first memory means, together with a trusted component which verifies the integrity and correct functioning of the computing platform. The trusted component comprises a second data processor and a second memory means, which are physically and logically distinct from the first data processor and first memory means.

[0043] In the example shown in Figs. 1 to 3 herein, the trusted computer entity is shown in the form of a personal computer suitable for domestic use or business use. However, it will be understood by those skilled in the art that that this is just one specific embodiment of the invention, and other embodiments of the invention may take the form of a palmtop computer, a laptop computer, a server-type computer, a mobile phone-type computer, or the like and the invention is limited only by the scope of the claims herein. In the best mode example described herein, the computer entity comprises a display monitor 100; a keyboard data entry means 101; a casing 102 comprising a motherboard on which is mounted a data processor; one or more data storage means e.g. hard disk drives; a dynamic random access memory; various input and output ports (not illustrated in Fig. 1); a smart card reader 103 for accepting a user's smart card; a confirmation key 104, which a user can activate when confirming a transaction via the trusted computer entity; and a pointing device, e.g. a mouse or trackball device 105. The trusted computer entity has a trusted component as described in the applicant's previous disclosure and as further described herein.

[0044] Referring to Fig. 2 herein, there are illustrated some of the components comprising the trusted computer entity, including keyboard 101, which incorporates confirmation key 104 and smart card reader 103; a main motherboard 200 on which is mounted first data processor 201 and trusted component 202, an example of a hard disc drive 203, and monitor 100. Additional components of the trusted computer entity, include an internal frame to the casing 102, housing one or more local area network (LAN) ports, one or more modem ports, one or more power supplies, cooling fans and the like are not shown in Fig. 2.

[0045] In the best mode herein, as illustrated in Fig. 3 herein, main motherboard 200 is manufactured comprising a processor 201; and a preferably permanently fixed trusted component 202; a memory device 300 local to the processor, the local memory device being a fast access memory area, e.g. a random access memory; a BIOS memory area 301; smart card interface 305; a plurality of control lines 302; a plurality of address lines 303; a confirmation key interface 306; and a data bus 304 connecting the processor 201, trusted component 202, memory area 300, BIOS memory area 301 and smart card interface 305. A hardware random number generator RNG 309 is also able to communicate with the processor 201 using the bus 304.

[0046] External to the motherboard and connected thereto by data bus 304 are provided the one or more hard disk drive memory devices 203, keyboard data entry device 101, pointing device 105, e.g. a mouse, trackball device or the like; monitor device 100; smart card reader device 103 for accepting a smart card device as described previously; the disk drive(s), keyboard, monitor, and pointing device being able to communicate with processor 201 via said data bus 304; and one or more peripheral devices 307, 308, for example a modem, printer scanner or other known peripheral device.

[0047] Smart card reader 103 is wired directly to smart card interface 305 on the motherboard and does not connect directly to data bus 304. Alternatively, smartcard reader 103 is connected directly to databus 304. To provide enhanced security confirmation key switch 104 is hard wired directly to confirmation key interface 306 on motherboard 200, which provides a direct signal input to trusted component 202 when confirmation key 104 is activated by a user such that a user activating the confirmation key sends a signal directly to the trusted component, by-passing the first data processor and first memory means of the computer platform.

[0048] Trusted component 202 is positioned logically and physically between monitor 100 and processor 201 of the computing platform, so that trusted component 202 has direct control over the views displayed on monitor 100 which cannot be interfered with by processor 201.

[0049] Confirmation key 104, and confirmation key driver 306 provide a protected communication path (PCP) between a user and the trusted component, which cannot be interfered with by processor 201, which by-passes data bus 304 and which is physically and logically unconnected to memory area 300 or hard disk drive memory device(s) 203.

[0050] The trusted component lends its identity and trusted processes to the computer platform and the trusted component has those properties by virtue of its tamper-resistance, resistance to forgery, and resistance to counterfeiting. Only selected entities with appropriate authentication mechanisms are able to influence the processes running inside the trusted component. Neither an ordinary user of the trusted computer entity, nor any ordinary user or any ordinary entity connected via a network to the computer entity may access or interfere with the processes running inside the trusted component. The trusted component has the property of being "inviolate".

[0051] The smart card may comprise a "cash card" or a "crypto card" the functions of which are described in the applicant's above-mentioned previous disclosure "Computing Apparatus and Methods of Operating Computing Apparatus", a copy of which is filed herewith, and the entire content of which is incorporated herein by reference.

[0052] On each individual smart card may be stored a corresponding respective image data which is different for each smart card. For user interactions with the trusted component, e.g. for a dialogue box monitor display generated by the trusted component, the trusted component takes the image data from the user's smart card, and uses this as a background to the dialogue box displayed on the monitor 100. Thus, the user has confidence that the dialogue box displayed on the monitor 100 is generated by the trusted component. The image data is preferably easily recognizable by a human being in a manner such that any forgeries would be immediately apparent visually to a user. For example, the image data may comprise a photograph 802 of a user. The image data on the smart card may be unique to a person using the smart card.

[0053] In the best mode herein, the trusted component operates to monitor data, including user data files and applications, on the computer platform by creating a set of data files which the trusted component dynamically monitors for any changes in the data, including absence of the data, which may occur as a result of the computer platform being compromised by a virus attack, or other interference. The trusted component is allocated or seizes a plurality of memory location addresses and/or file directories in the first memory area of the computer platform, which become a user space reserved for use by the trusted component. The reserved memory area comprises a selected proportion of the total memory area of the computer platform. Within the reserved memory area, the trusted component preferably creates a plurality of directories. Within the reserved memory area, the trusted component also creates a plurality of data files, which can either be copies from real user files on the computer platform, or which can be created by the trusted component. The primary purpose of these files is to act as a set of files to which the trusted component has access, and to which ordinary user accounts of the computer platform, under normal operation, do not have access. Because the files in the reserved memory area are reserved for use by the trusted component and under normal operation, are not accessed by applications on the computer platform, the trusted component can use files stored in the reserve memory area as a "control" set of files by which to monitor for unauthorized changes to the data, for example as caused by a virus. It will be appreciated that the 'NT administrator' or the 'Unix super user' or similar accounts with overriding powers must refrain from making changes to the data in the reserved space, even though it can.

[0054] Because the files stored in the reserve memory area are either duplicates of user files, duplicates of applications or files created specifically by the trusted component, they are of little or no value to the computer platform for performing its normal operating functions. If the files in the reserve memory area become corrupted for any reason, they may be sacrificed and are easily replaceable. However, since access to the reserved access memory area is restricted to the trusted component, any corruption of the files in the reserve memory area is deemed to be indicative of changes to files occurring through undesirable mechanisms, e.g. by a virus program. The files in the reserve memory area are periodically monitored by the trusted component to check for such corruption. If corruption is discovered, by the monitoring process, then a measure of the likely corruption of the remaining memory area on the computer platform can be determined by probabilistic methods.

[0055] By providing a reserved memory area containing files which can be sacrificed, if the computer platform is compromised by a hostile attack, e.g. a virus, then the sacrificial files stored in the reserved memory area are at least as likely to be affected as other user data files stored in the remaining portion of the memory of the computer platform. Thus any corruption of the files in the reserve memory area, if spotted early enough, may give an indication to the trusted component that file corruption is occurring on the computer platform, in which case the trusted component can take action to limit the spread of corruption at an early stage, and preferably before damage is done to important data files stored in the remaining memory area of the computer platform.

[0056] Referring to Fig. 4 herein, there is illustrated schematically an internal architecture of trusted component 202. The trusted component comprises a processor 400, a volatile memory area 401; a non-volatile memory area 402; a memory area storing native code 403; and a memory area storing one or a plurality of cryptographic functions, 404, the non-volatile memory 401, native code memory 403 and cryptographic memory 404 collectively comprising the second memory means hereinbefore referred to. The cryptographic functions 404 preferably include a source of random numbers.

[0057] Trusted component 202 comprises a completely independent computing entity from the computer platform. In the best mode herein, the trusted component shares a motherboard with the computer platform so that the trusted component is physically linked to the computer platform. In the best mode, the trusted component is physically distinct from the computer platform, that is to say it does not exist solely as a sub-functionality of the data processor and memory means comprising the computer platform, but exists separately as a separate physical data processor 400 and separate physical memory area 401, 402, 403, 404. By providing a physically present trusted component, the trusted component becomes harder to mimic or forge through software introduced onto the computer platform. Programs within the trusted component are pre-loaded at manufacture of the trusted component, and managed by the manufacturer of the trusted component, and are not user configurable. The physicality of the trusted component, and the fact that the user component is not configurable by the user enables the user to have confidence in the inherent integrity of the trusted component, and therefore a high degree of "trust" in the operation and presence of the trusted component on the computer platform.

[0058] Referring to Fig. 5 herein, there is illustrated schematically a logical architecture of the trusted computer entity referred to in Figs. 1 to 4 herein. The logical architecture has a same basic division between the computer platform, and the trusted component, as is present with the physical architecture described in Figs. 1 to 3 herein. That is to say, the trusted component is logically distinct from the computer platform to which it is physically related. In Fig. 5, components of the computer entity are illustrated within the dotted line 500, whereas elements external to the computer entity, such as the Internet 501, and one or more local area networks or wide area networks 502, and a user's smart card 503 are shown outside dotted line 500. Logically, the computer entity is divided into "user space" 504, comprising all logical areas accessible by computer platform, and "trusted component space" 513 comprising areas accessible solely by the trusted component 203. The user space 504 includes one or more communications port facilities 506; one or more applications programs 507, for example a word processor, database package, accounts package, Internet access application, etc; a set of file directories 508; smart card interface 305, for interfacing with the smart card reader 103, optionally a random number generator 309, and optionally, a software agent 511 which is used by the trusted component to manipulate files and applications in user space, and to report back to the trusted component. Optionally a software agent 514 is used by the smartcard 503 to manipulate files and applications in user space, and to report back to the smartcard.

[0059] In the trusted component space, are resident the trusted component itself, displays generated by the trusted component on monitor 100; and confirmation key 104, inputting a confirmation signal via confirmation key interface 306.

[0060] Within the file directory area 508 is a set of reserved directories 512 for creation of a set of data files reserved for use by the trusted component, and used for monitoring in the user space according to the specific methods described herein. For ease of reference, such files will hereafter be referred as 'reserved files'.

[0061] The random data generator 309 is used to generate random data, forming the content of various of the reserved files in the reserved one or more directories.

[0062] Referring to Fig. 6 herein, the software agents 511 and 514 each comprise a corresponding respective one or more file manipulation programs 600; and a corresponding respective communications interface 601. The file manipulation program(s) within the software agent 514 in user space operate on instruction from smartcard 503 to: monitor a plurality of said data files in the one or plurality of directories in user space reserved for use by the user of the smartcard; copy said files to user space reserved for use by the user of the smartcard but which also allows read access by the trusted component; and delete said copied files. The file manipulation program(s) within the software agent 511 in user space operate on instruction from trusted component 202 to: create and monitor a plurality of data files in the one or plurality of directories in user space reserved for use by the trusted component; copy files from user space reserved for use by the user of the smartcard but which also allows read access by the trusted component.

[0063] Referring to Fig. 7 herein, there is illustrated schematically logical components of trusted component 202. Trusted component 202 comprises a communications interface 700; a set of cryptographic functions 701 including a random number generator and cryptographic algorithms for communicating with smart card reader 103; one or more monitoring applications 702 for monitoring data relating to reserved files; a display interface program 703 for generating an interactive display on monitor 100 and allowing interface via the display using pointing device 105 and keypad 101; optionally one or more file manipulation programs 704; native code 705 for monitoring files by gathering and reporting information describing the data content of the files to the trusted component as in software agent 511; and a dictionary program 706 for generating text and strings of text for using as data to name directories and files and fill reserved files in user space 504. The trusted component also contains a dictionary of file names, which is used automatically to name and rename the reserved file directories and reserved files.

[0064] There will now be described specific methods of operation of the computer entity for security monitoring of data files in the computer platform, by the trusted component 202. In the following description, there are illustrated in Figs. 8-12 herein main process steps operated by the trusted component and computer platform for performing the method. It will be understood by those skilled in the art that such process steps may be implemented in the form of code in a conventional programming language stored in the memory of the trusted component and the computer platform. Steps relating to operations carried out in user space, in general are executed on the processor 201 according to code stored in either the memory of the trusted component or the computer platform, but some part(s) of those operations may be carried out inside the trusted component 202 according to code stored in a memory device of the trusted component. On the other hand, where process steps are shown as operating in trusted component space, the steps are executed within the trusted component 202 (for example on the processor 400) according to code stored in a memory device of the trusted component. Implementation of such code is well known by those skilled in the art, and such code may be written in conventional computer program languages such as C, C++, or the like.

[0065] Referring to Fig. 8 and 9 herein, there will now be described a first mode of operation for a security monitoring process, which is activated by user. The computer platform generates a conventional operating system display having a plurality of icons, for example a display as produced by the Windows 95™ operating system. An icon is provided on the operating system display, created in known manner, which when activated by a user using the pointing device, e.g. mouse 105 results in a dialog box display generated by trusted component 202, for example as shown in Fig. 8 herein. The dialog box 800 is generated by display interface 703 of trusted component 202. The dialog box 800 comprises one or more menu displays 801, displayed upon a display background comprising an image 802 retrieved from a user smart card 503 which must be inserted into smart card reader device 103 in order to provide the image on the dialog box display background. Since the image displayed on the dialog box background is that stored on the user smart card, the user can be confident that the dialog box is created by the trusted component, since the obtaining of the image data from the smart card is carried out by obtained encrypted image data from the smart card using crypto functions 404, 701 stored in the trusted component. On viewing the dialog box 800, the user may activate pointing device 105 to click on an icon display 803 to produce a drop-down menu 801 with options for file manipulation. For example, menu options may include icons to: start file monitoring; stop file monitoring; enable the copying of files; disable the copying of files; enable the creation of encrypted files; disable the creation of encrypted files; delete files in the reserved memory area of the computer platform or to display metrics of files. On the user selecting one of the options, the trusted component generates a confirmation display 900 prompting the user to activate the confirmation key 104. Because the confirmation key 104 is wired directly to the trusted component 202, activation of the confirmation key provides a secure method by which the trusted component is activated directly by the user without any third party intervention, and ensures that the options selected through the menu display from the pointing device input 105 are independently confirmed by separate key activation through a separate channel avoiding data bus 304, and avoiding the computer platform, and directly to the trusted component 202.

[0066] Referring to Fig. 10 herein, there is illustrated schematically process steps operated by the combination of the software agent 511 and trusted component 202 for monitoring of data files file directories 508.

[0067] In step 1000, the trusted component 202 seizes a portion of the memory capacity of the computer platform, for example hard disc or RAM for exclusive access by the trusted component, via the software agent 511. The software agent 511 may seize the memory area by creating one or a plurality of directories, for its own use, either directly, bypassing the operating system functions for file creation, or alternatively by making appropriate instructions to the operating system to create the appropriate directory or directories. Agent 511 creates a plurality of data files in those reserved directories in step 1001. Creation of data files can be by three methods. Firstly, file creation may occur by the copying into the reserved directories of existing files on the computer platform belonging to the user, with the user's permission. Secondly, agent 511 may allocate file names within those reserved directories. The file names and names of reserved dictionaries being provided by dictionary program 706. The data within the files is provided by dictionary program 706 within the trusted component 202 which generates individual words and strings of words of text which are passed to agent 511, which then writes those words or strings of words into the created reserved files in the reserved directories. Thirdly, the agent 511 may create its own data files of substantially random data, by storing random bits generated by random number generator 309 (or by the random number generator inside the trusted component's cryptographic functions 404) in the created files. In step 1002, agent 511 monitors the plurality of created reserved data files stored in the reserved memory area 512. A data digest of each memory file created by agent 511 is produced by applying a hash function algorithm to the data. The hash function may be applied by the agent 511 and the digest data for each agent created file reported back periodically to trusted component 202, which stores the digest data in its trusted memory area 402. Alternatively, the agent 511 may periodically report each agent created file to the trusted component 202, which generates its own digest using its crypto functions 404 and stores the digest data in its trusted memory area 402. Trusted component 202 stores at least two digest data, comprising a previous digest data and a most recently received current digest data for each monitored reserved data file in its memory area 402. Trusted component 202 operates an algorithm comprising digest monitoring component 702, to check whether the previous digest data of each particular agent created data file is identical to the current digest data of that file. Within digest monitoring component 702, there is provided a separate file space into which results of the monitoring process are recorded. Any changes to the reserved data files in the reserved memory area in user space 508 discovered by monitoring the digest data within the trusted component are recorded to the error record file within the trusted component in step 1003. From time to time changes in data files stored in the reserved memory area may occur due to normal system malfunctions, which are not due to hostile attack by external stimuli, e.g. viruses. However, such changes to the data files may be very rare. An algorithm within the monitoring component of 702 of the trusted component applies a statistical test as to whether any changes to data files which have been recorded in the error file are statistically relevant. For example, the algorithm within the trusted component may be preset to allow a predetermined number of errors to occur within any given period. For example, an error level of one error per month on a predetermined number of reserved files may be preset as an allowable rate of errors. If more errors occur than this in the predetermined time, giving rise to a significant level of errors in the monitored files in the reserved memory area, in step 1004, the test applied by the trusted component to see whether such tests are significant may prove positive. If no significant changes in the stored data files are determined in step 1005, the trusted component and agent 511 continues to periodically monitor the selected data files in the reserved area of user memory on the computer platform in step 1002. If the number of errors are significant, in step 1006 the trusted component may generate an alarm data indicating a significant level of data corruption in the monitored files, and in step 1007 may report such corruption by generating a display on monitor 100. Further, on experiencing a significant level of errors in the monitored data files, resulting in an alarm condition, the trusted component may notify any other third party entities communicating with the computer platform, that the computer platform is in an alarm condition and possibly that data on the platform or the functionality of the platform has been compromised.

[0068] Applying a hash program to data in the user space 504 using the main processor 201 and sending the digest to the trusted component 202 is fast, because of the superior processing capabilities of user space 504, but has the disadvantage that the hash program may have been subverted (by a virus, for example), so there is a reduced level of confidence in the digest given to the trusted component. Sending the entire original file data to the trusted component, and causing the trusted component to compute a digest using its own resources, for example on processor 400 and crypto functions 404, has the disadvantage that the process is slower, because the trusted component has inferior computing capability than the user space 504. It has the advantage that the hash program cannot be subverted, hence there is greater confidence in the value of the digest.

[0069] Where file manipulation is carried out by agent 511, its file manipulation program 600 runs continuously, monitoring files in the reserved directories, and reporting back to trusted component 202.

[0070] Because the files in the reserved directories in the user space which are created by agent 511, look to a computer virus as being exactly the same as real data files, or in the case of random data, look the same as encrypted data files, a hostile virus having entered into the computer platform is equally likely to affect the sacrificial files stored in the reserved directories as it is to attack user data files in the user space. The proportion of data files, in terms of file numbers, or in terms of megabytes of data stored in the files, can be selected by a user, by means of a drop-down menu in a dialog box, or can be preset in the trusted component, or agent software. For example, if the number of sacrificial files is set at being 10% of the number of real user files created, then for viruses which identify files by file name, there is a corresponding percentage (for example 10%) probability that a virus will attack the sacrificial files in reserved directories, of all the files including sacrificial files and user files stored on the computer platform. Thus, the extent, number and frequency of errors occurring in the sacrificial file directories may give a statistical measure of the extent of damage done by a computer virus. Because monitoring of the sacrificial files is continuous, whilst the computer entity is operating, the best mode herein may provide real time monitoring for attack by external programs, which provides an alarm function to operate when a predetermined level of file corruption has occurred on the computer platform, in a manner which cannot be interfered with by users of the computer platform, thereby lending increased trustability to the computer entity.

[0071] Referring to Fig. 11 herein, there is illustrated schematically in more detail process steps carried out by agent 511 in creating data files in the reserved directories by copying user files, and reporting to trusted component 202. In step 1100, agent 511 obtains a named user data file from the untrusted memory area of the computer platform, with the user's permission. The user gave that permission by selecting the 'enable copy files' option on the file menu 801 in the security function dialog box 800. The user may indicate his or her permission for copying the file, by pressing confirmation key 104 upon display of the pressed confirmation key display prompt 900 as described previously. In step 1101, agent 511 selects a new file name for the file copied over from the user files. The new file name may be selected randomly by agent 511, or may be generated according to predetermined rules stored in agent 511. A different file name is created, for the copied user file in the reserved directory in which it is stored, which is specific to the agent 511 and accessible only by agent 511. In step 1102, the agent stores the renamed user data file in the reserved directory in the reserved memory area of the (untrusted) computer platform memory area, e.g. the hard disc or RAM. In step 1103, agent 511 applies a hash function to the data file to generate a corresponding respective digest data for the file. The digest data is reported back to trusted component 202 in step 1104. The trusted component stores the digest data in a trusted memory area 402 as described previously in step 1105. In step 1106, the trusted component determines whether it is the first time that a digest data has been created for that file. That is to say, the trusted component determines whether an historical data for that particular file already exists in the trusted component's memory area 402, or whether the currently obtained digest data from agent 511 is the first data obtained for that file. If the digest data obtained for that file is the first digest data obtained for that file, then the trusted component stores the digest data in its trusted memory area as current digest data, and waits for agent 511 to report a further digest data on the same file after a predetermined monitoring period i.e. waits until agent 511 applies a hash function as per step 1103 described previously and reports a new current digest data. On receiving the new current digest data (the second digest data for that file) the trusted component then has a current and an historical digest data for that file, and can make a comparison between the current and historical stored digest data in trusted memory area 402 for a particular file in step 1107. If the result of the comparison is that the current digest data for a particular file is the same as previous historical digest data for the file in step 1108, then after waiting a predetermined period in step 1109 during which agent 511 periodically monitors the user data file, agent 511 applies the hash function in step 1303 and reports the digest data to trusted component 202 in steps 1103-1104. However, if it is determined that there is a change in the current digest data for a particular file compared to the previously reported historical digest data in step 1108, then the trusted component records the details of the file number and time of the change in data in the error file in the trusted memory area 402 for that file in step 1110.

[0072] Hash functions are well-known in the prior art and comprise one way functions which are capable of generating a relatively small output data from a relatively large quantity of input data, where a small change in the input data results in a significant change in the output data. Thus, a data file to which is applied a hash function results in a first digest data (the output of the hash function). A small change e.g. a single bit of data in the original data file will result in a significantly different output when the hash function is reapplied to the modified data file. Thus, a data file comprising megabytes of data may be input into the hash function and result in a digital output of the order of 128 to 160 bits length, as the resultant digest data. Having a relatively small amount of digest data generated from a data file stored in the reserved directory is an advantage, since it takes up less memory space and less processing power in the trusted component.

[0073] Referring to Fig. 12 herein, there is illustrated schematically interactive steps between trusted component 202, proxy agent 511, proxy agent 514, and smart card 503, via smart card interface 305 and smart card reader 103 for copying of user files to reserved directories, and for continuous monitoring of sacrificial reserved files in the user space 504.

[0074] In step 1200, trusted component 202 requests the smart card, via agent 511, for an update of a user's data files. In step 1201, smart card 503 receives the request, which at this stage is unauthorized, and in response to receipt, in step 1202 sends a nonce to the agent 511 which is received by the agent in step 1203. The nonce may comprise a string of random bits created by smart card 503. The agent concatenates the request and the nonce, signs the concatenation, and sends the request and nonce and signature back to the smart card which is received by the smart card in step 1205 so that the smart card can verify that the trusted component is on-line. Smart card 503 uses its proxy agent 514 operating in user space on behalf of the smartcard and/or a pre-stored program on the smart card to make an inventory of the user's files and sends the inventory back to the trusted component in step 1208, after first verifying the request in step 1206 and constructing a file inventory which to send, in step 1207. The file inventory is received by agent 511 in step 1209. The trusted component 202 or the agent 511 uses the information on the file inventory by operating an algorithm to identify new or altered user files, and creates new directories in the reserved user space directories 512 allocated to the trusted component. The trusted component in step 1210 requests from the smart card or its proxy agent 514 copies of the new user files, and the smart card in step 1211 receives the request in step 1212. The smartcard or its proxy agent 514 copies the named user files into a memory space where the trusted component has read access, and then indicates in step 1213 that the copied files are ready for monitoring. In step 1214, the agent 511 ensures that the files are copied from the user space to the reserved directories allocated to the trusted component. The file names are renamed in step 1215, as previously described with reference to Fig. 11, and when agent 511 indicates that the files have been renamed, the smartcard and/or its agent 514 deletes the copied files from the memory space where the trusted component has read access. At this stage, files have been copied from user space to the reserved directories allocated to the trusted component, and then in step 1217 further read access to the agent 511 is denied by smart card 503. The agent then continues in step 1218 to compute file digests by applying the hash function to each individual file in the reserved directories in user space, which are then reported to the trusted component periodically as described previously. In step 1219 the trusted component stores the digests inside the trusted component, and generates the necessary error records if errors occur and generates alarms and reports to the monitor 100 and other entities as described previously herein with reference to Figs. 10 and 11.

[0075] The file manipulation program 600 may optionally be stored within the trusted component as file manipulation program 704, so that instead of the agent 511 corresponding with the smart card and computer platform memory for copying and monitoring of files, this may be done from inside the trusted component in a variation of the embodiment.

[0076] Since the best mode herein operates by monitoring the data on a computer platform, there may be provided a system which is immune to variations of virus programs and new generations of viruses, but which is capable of detecting the effects of any virus which operates by changing data on a computer platform.

Claims

1. A method of security monitoring of a computer platform, said method comprising the steps of:

   (i) creating a data file in a memory area of said computing platform;
   (vi) generating a first digest data describing a data content of said data file;
   (vii) waiting a predetermined time period;
   (viii) repeating step (ii) to generate a second digest data; and
   (ix) comparing said second digest data with said first digest data.

2. The method as claimed in claim 1, further comprising the step of:
   (vi) if said second digest data is identical to said first digest data, repeating steps (ii) to (v) above.

3. The method as claimed in claim 1, further comprising the step of:
   (vii) if said second digest data is not identical to said first digest data, storing an error data in said trusted memory area.

4. The method as claimed in claim 1, wherein said step of generating a first digest data comprises applying a hash function to said data file to produce a hash function data corresponding to said data file.

5. The method as claimed in claim 1, wherein said step of creating a data file in a memory area of said computer platform comprises copying an existing user data file into a reserved portion of said memory area of said computer platform.

6. The method as claimed in claim 1, wherein said step of creating a data file in said memory area comprises generating a random or pseudo random data in a reserved portion of said memory area of said computer platform.

7. The method as claimed in claim 1, wherein said step of generating a digest data corresponding to said data file is carried out by an algorithm operating on said computer platform.

8. The method as claimed in claim 1, wherein said step of generating a digest data comprises sending a said data file to a trusted component comprising a trusted processor and a trusted memory area, and generating said digest data by applying an algorithm to said data file in said trusted component.

9. A computer entity comprising:

   a computer platform comprising a first data processing means and a first memory means;

   a monitoring component comprising a second data processing means and a second memory means;

   wherein said monitoring component comprises means for receiving a monitor data, said monitor data describing a content of a plurality of data files stored in said computer platform in said first memory means;

   means for storing said plurality of monitor data in said monitoring component; and

   means for making comparisons of said monitor data,

   wherein said monitoring component receives for each of a plurality of data files, an historical monitor data representing a state of said data file at a previous point in time, and a current monitor data representing a current state of said data file.

10. The computer entity as claimed in claim 9, wherein said historical monitor data and said current monitor data are stored in said second memory means of said monitoring component.

11. The computer entity as claimed in claim 9, wherein said monitoring component comprises a set of agent code stored in said second data storage means, wherein said set of agent code may be transferred to said first data storage means for operation and control by said first data processing means in said computer platform.

12. The computer entity as claimed in claim 9, wherein said monitoring component comprises a dictionary means, said dictionary means comprising a text generator device operable to generate a plurality of text data representing a plurality of words, and said monitoring means transferring said text data to a plurality of said data files created in a reserved area of said first memory means.

13. The computer entity as claimed in claim 12, wherein said dictionary means is operable to generate a plurality of names identifying said plurality of data files created in said reserved area of said first memory means.

14. The computer entity as claimed in claim 12, wherein said dictionary means is operable to generate a plurality of names of directories containing said plurality of data files created in said reserved area of said

first memory means.

15. The computer entity as claimed in claim 9, further comprising an agent means, said agent means resident and operating on said computer platform wherein,

said agent means creates a plurality of said data files in a reserved region of said first memory area;

said agent means substantially continuously monitors said created data files in said reserved memory region; and

said agent reports said monitor data describing a content of said data files in said reserved memory region periodically to said monitoring component.

16. The computer entity as claimed in claim 9, comprising a random data generator. wherein said random data generator generates random data which is stored in a plurality of said data files created in a reserved region of said first memory area of said computer platform.

17. The computer entity as claimed in claim 9, comprising an agent device resident on said computer platform, and a smart card reader device, wherein said agent device communicates with said smart card reader device for receiving a file name data from said smart card reader device, said file name data describing one or a plurality of file names of user data files for which permission to copy said user data files is granted to said agent device.

18. A method of security monitoring a computer platform comprising a first data processing means and a first memory means, said method comprising the steps of:

receiving a first monitor data, said first monitor data describing a data content of a data file stored in said computer platform;

storing said first monitor data in a trusted memory area physically and logically distinct from said computer platform;

receiving a second monitor data, said second monitor data describing a data content of said same data file stored in said computer platform;

comparing said first monitor data with said second monitor data; and

if said first monitor data differs from said second monitor data, generating an error data.

19. The method as claimed in claim 18, further comprising the step of generating said first monitor data by applying a one-way function algorithm to a data content of said data file.

20. The method as claimed in claim 18, further comprising the step of:

generating an alarm display when a said error data is created.

21. The method as claimed in claim 18, further comprising the step of:

comparing said error data against a predetermined measure of error data allowable in a predetermined time, to determine if said error data is statistically significant.

22. The method as claimed in claim 21, further comprising the step of:

if said error data is determined to be statistically significant, generating an alarm display indicating an error has occurred in said data file.

23. A method of monitoring a computer platform comprising a first data processing means and first memory means, said method comprising the steps of:

a) allocating a region of said first memory means for use by a monitoring entity comprising a second data processing means and a second memory means;

b) creating in said allocated memory area a plurality of data files, each allocated to said monitoring entity;

c) entering data into said plurality of allocated data files in said reserved memory region;

d) creating for each of said data files a monitor data describing a data content of each of said data file;

e) storing said monitor data in a second memory device, said second memory device being physically and logically distinct from said first memory device;

f) repeating steps d) and e); and

g) periodically comparing a recently received said monitor data for said data file with a previously received monitor data for the same said data file.

24. The method as claimed in claim 23, wherein said step of entering data into a said data file comprises:

copying an existing data file from an unreserved area of said first memory device into said data file.

25. The method as claimed in claim 24, further comprising the step of:

assigning a file name of said data file in said reserved memory area said file name being a different file name to a file name of said original user file from said unreserved area of said first memory area from which said data file was copied.

26. The method as claimed in claim 24, further comprising the step of:

assigning a directory name of a directory used for storing said data file in said reserved memory area said directory name being a different directory name to a directory name of said original user directory from said unreserved area of said first memory area in which said data file was originally located.

27. The method as claimed in claim 23, wherein said step of creating a monitor data comprises:

applying a one-way function algorithm to data in said data file, to produce said monitor data from said data stored in said data file.

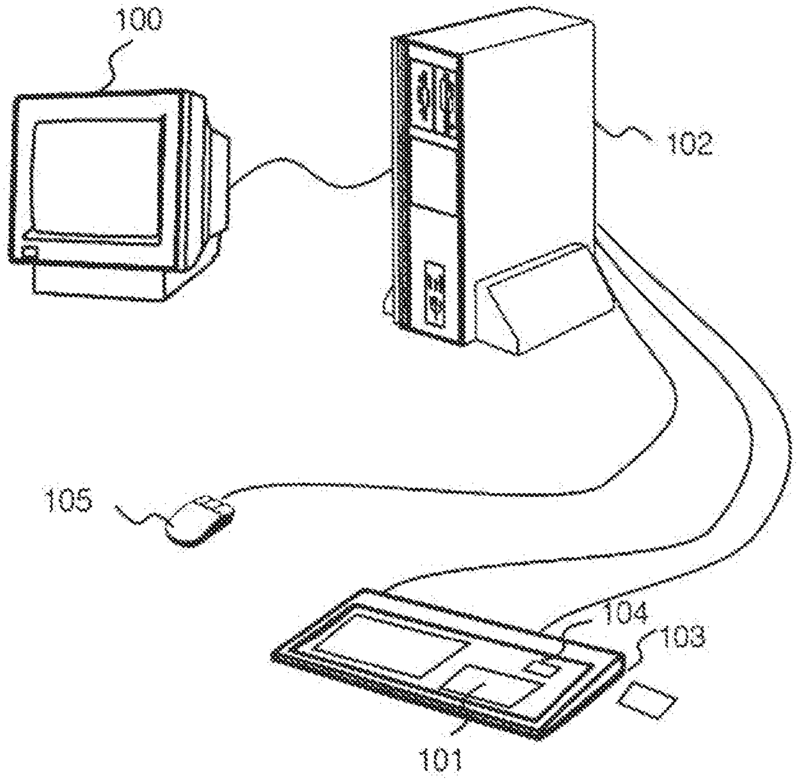


Fig. 1

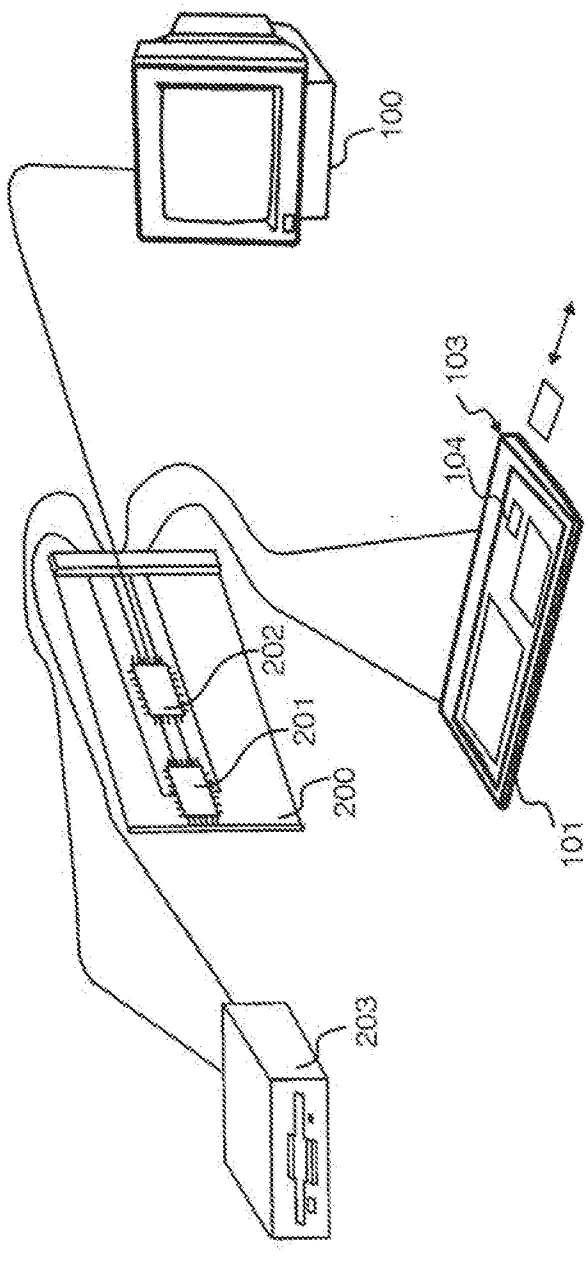


Fig. 2

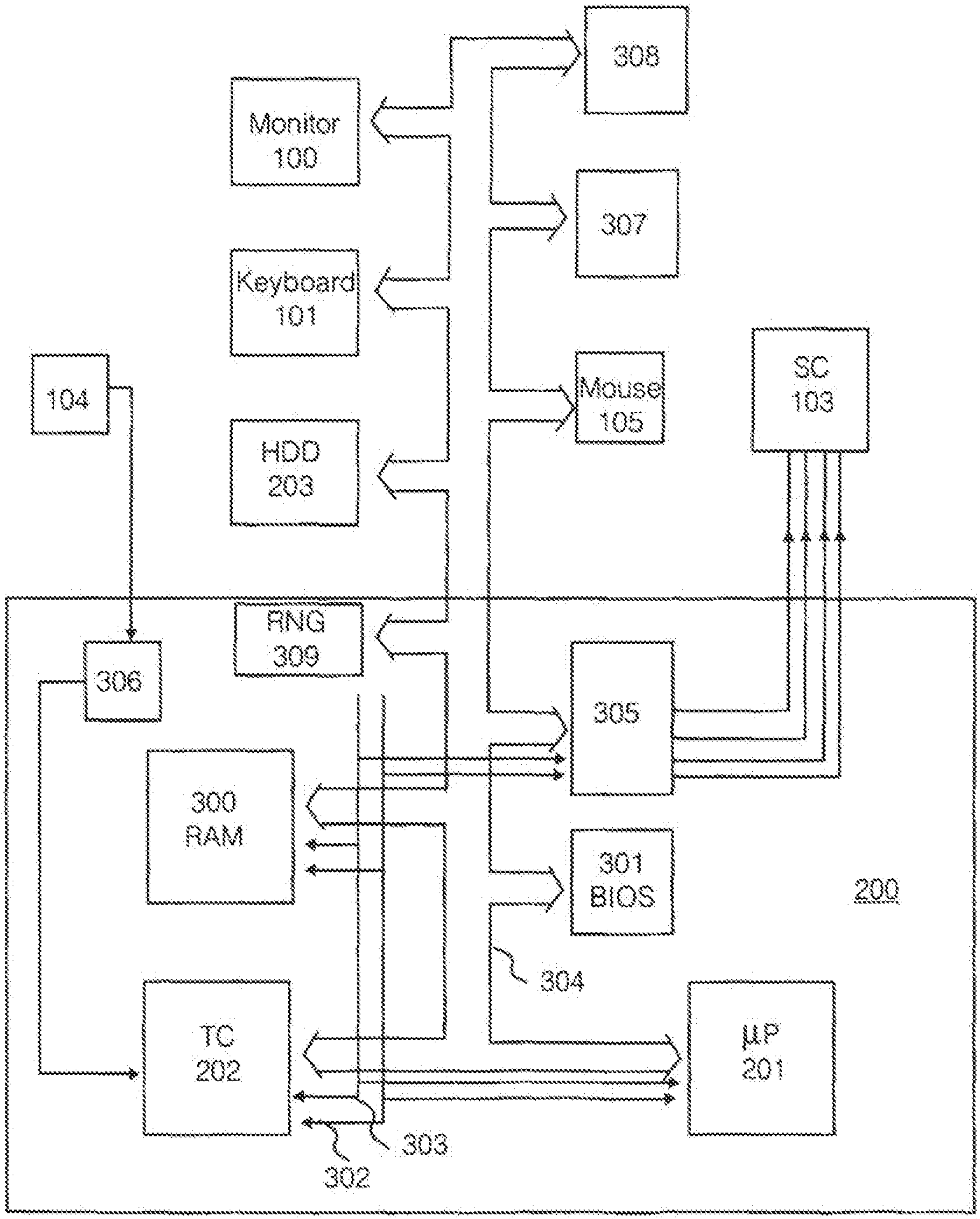

308
Monitor
100
307
Keyboard
101
SC
103
104
Mouse
105
HDD
203
RNG
309
306
305
300
RAM
301
BIOS
200
304
TC
202
µP
201
303
302

Fig. 3

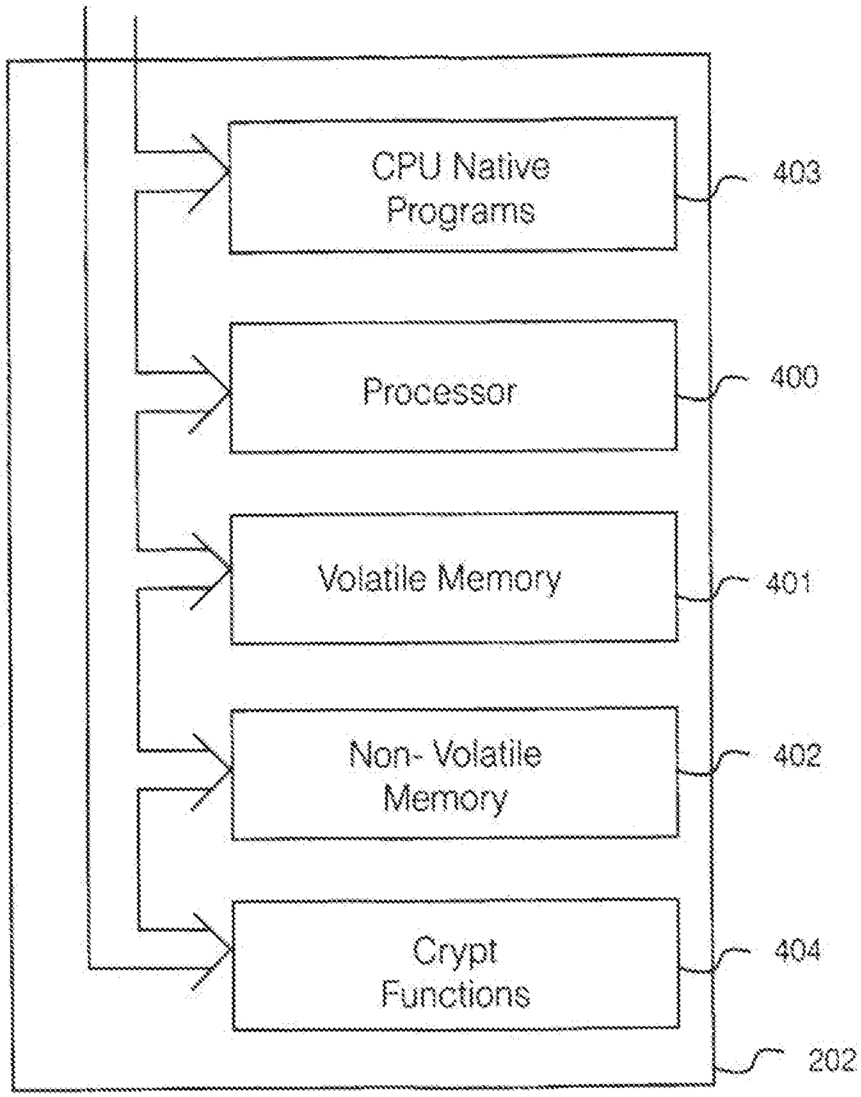

CPU Native Programs
403
Processor
400
Volatile Memory
401
Non- Volatile Memory
402
Crypt Functions
404
202

Fig. 4

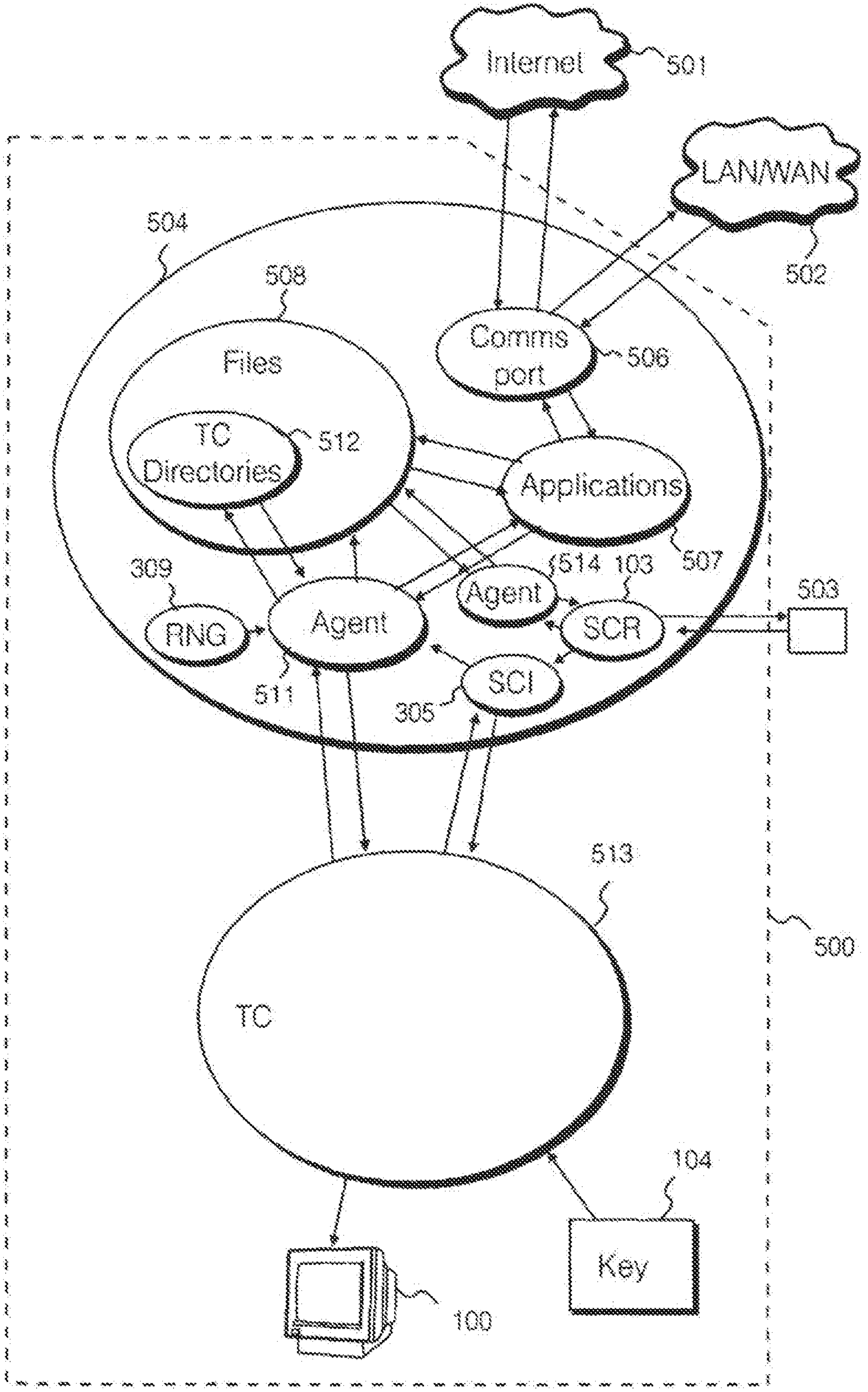

Internet
501
LAN/WAN
502
504
508
Files
Comms port
506
TC Directories
512
Applications
507
309
514
103
503
RNG
Agent
Agent
SCR
511
SCI
305
513
TC
500
104
Key
100

Fig. 5

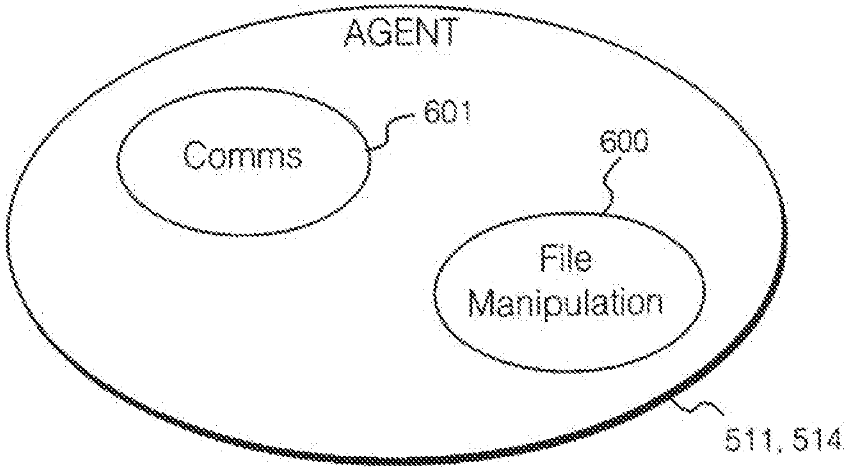

AGENT
Comms
601
600
File Manipulation
511, 514

Fig. 6

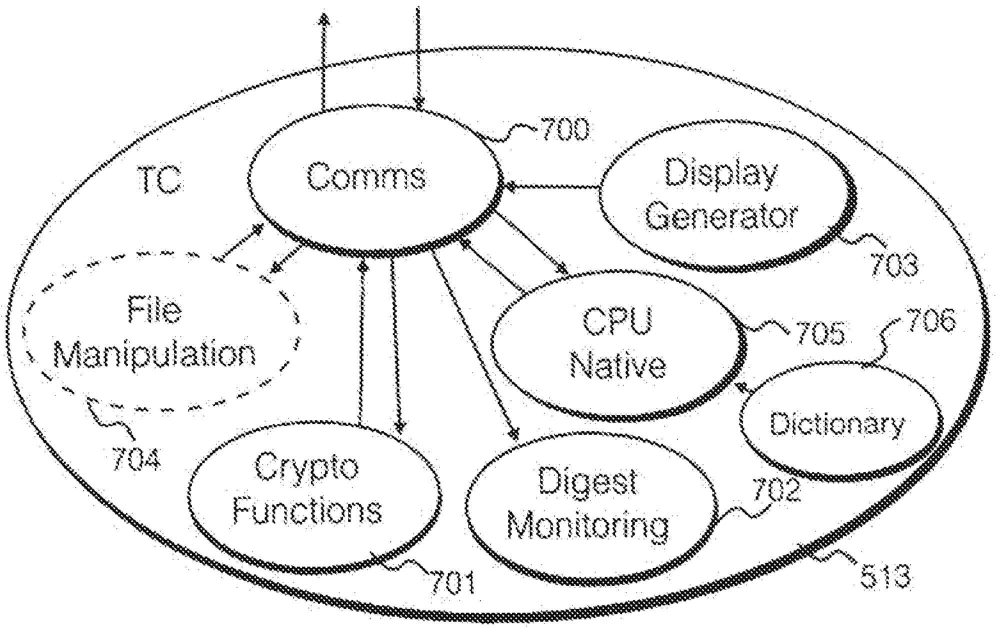

TC
Comms
700
Display Generator
703
File Manipulation
704
CPU Native
705
706
Dictionary
Crypto Functions
701
Digest Monitoring
702
513

Fig. 7

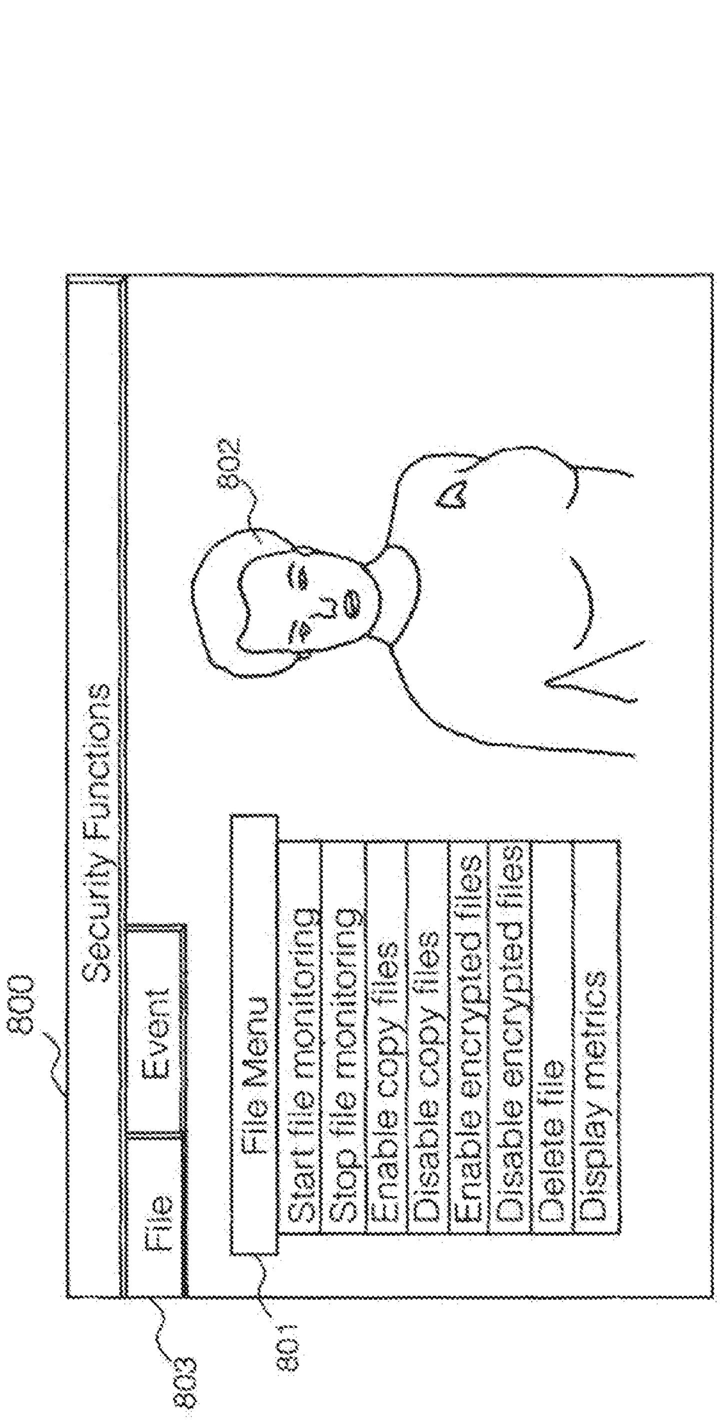
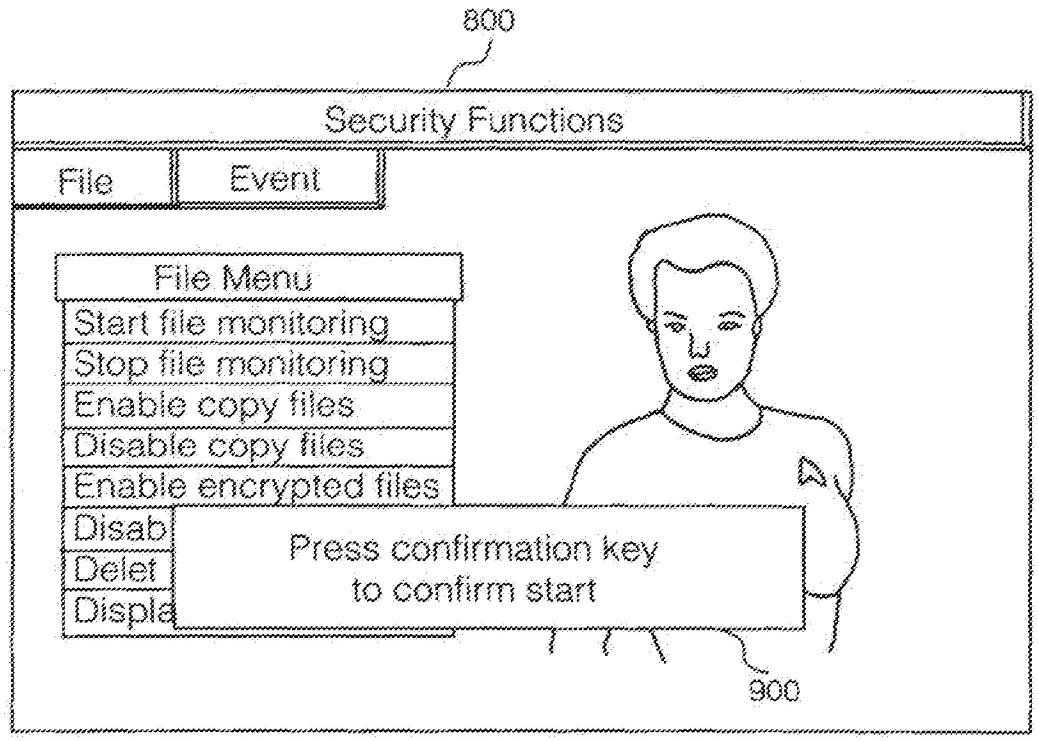

800
Security Functions
File
Event
803
File Menu
Start file monitoring
Stop file monitoring
Enable copy files
Disable copy files
Enable encrypted files
Disable encrypted files
Delete file
Display metrics
801
802

Fig. 8



Press confirmation key
to confirm start

Fig. 9

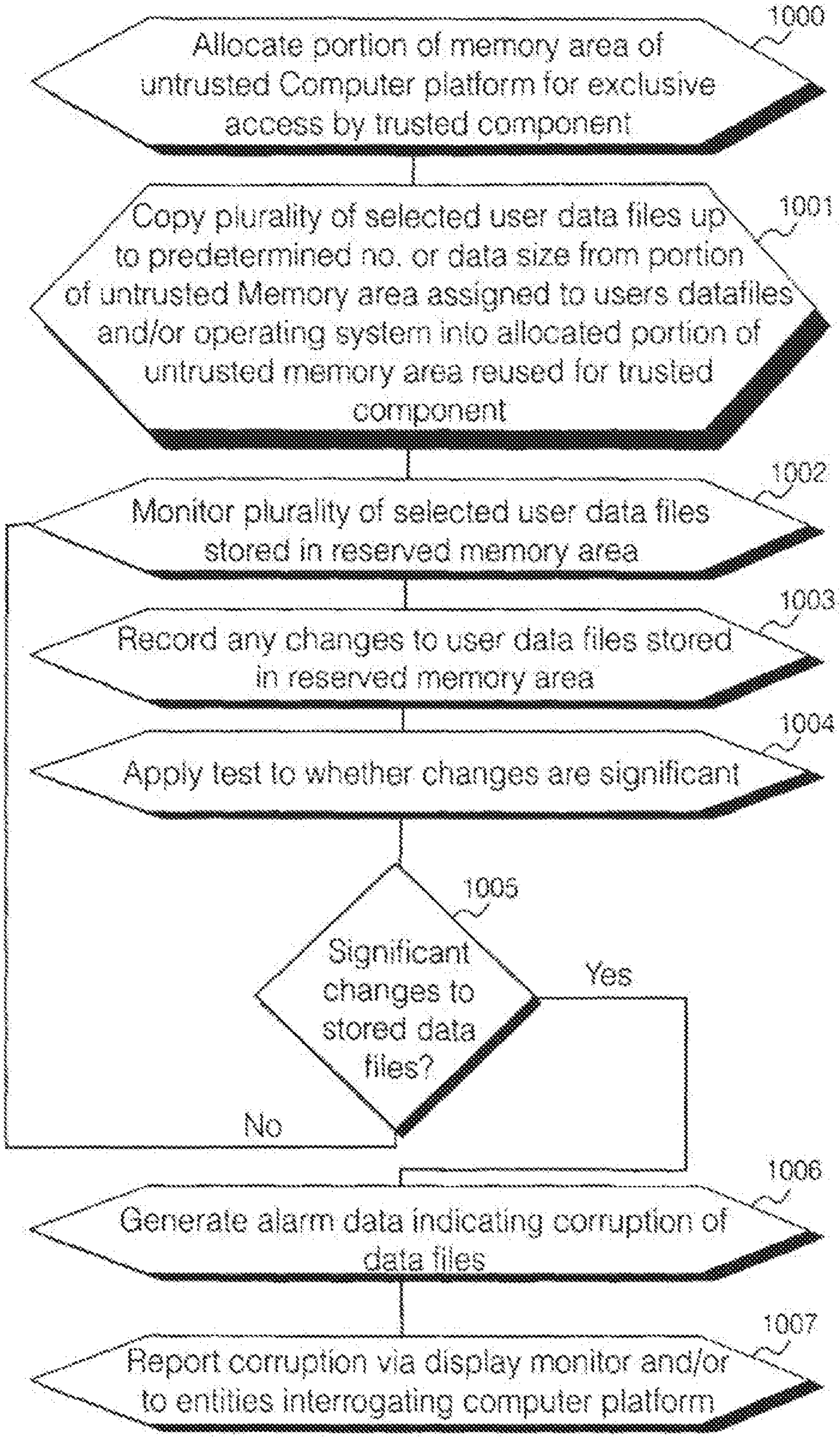

1000
Allocate portion of memory area of untrusted Computer platform for exclusive access by trusted component
1001
Copy plurality of selected user data files up to predetermined no. or data size from portion of untrusted Memory area assigned to users datafiles and/or operating system into allocated portion of untrusted memory area reused for trusted component
1002
Monitor plurality of selected user data files stored in reserved memory area
1003
Record any changes to user data files stored in reserved memory area
1004
Apply test to whether changes are significant
1005
Significant changes to stored data files?
Yes
No
1006
Generate alarm data indicating corruption of data files
1007
Report corruption via display monitor and/or to entities interrogating computer platform

Fig. 10

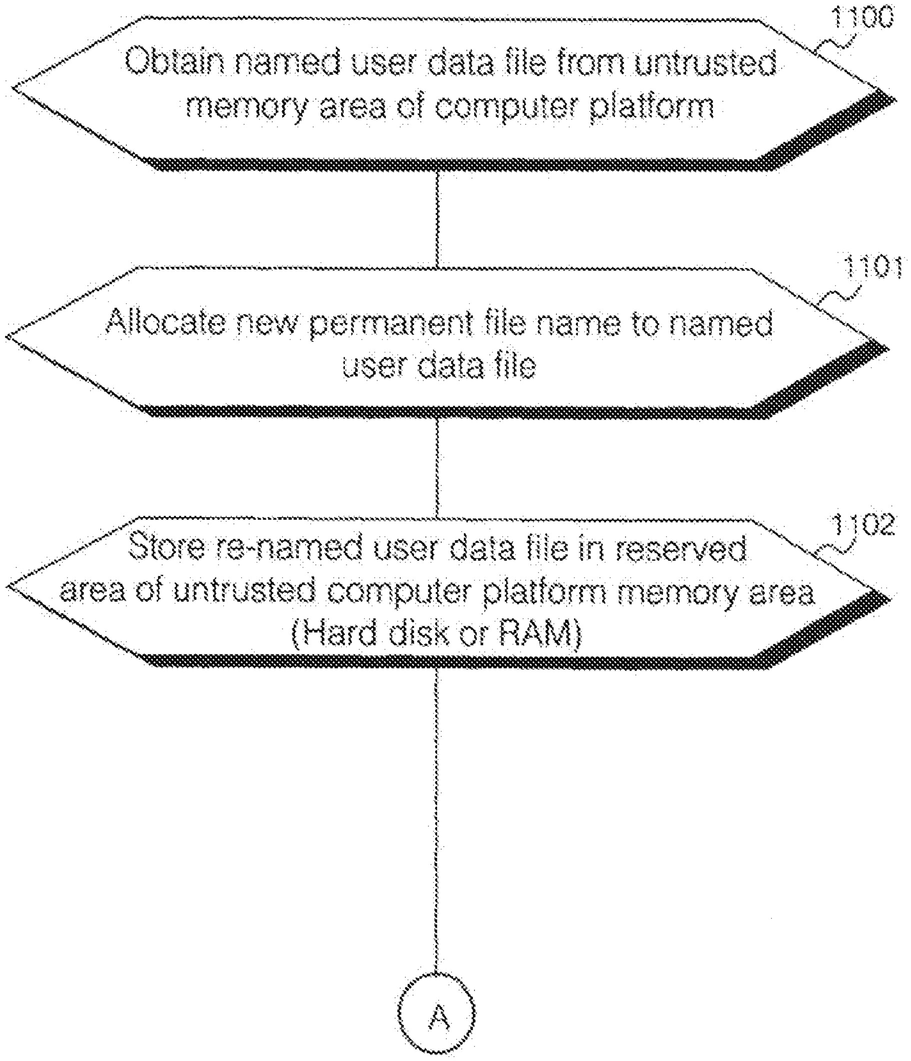

1100
Obtain named user data file from untrusted memory area of computer platform
1101
Allocate new permanent file name to named user data file
1102
Store re-named user data file in reserved area of untrusted computer platform memory area (Hard disk or RAM)
A

Fig. 11

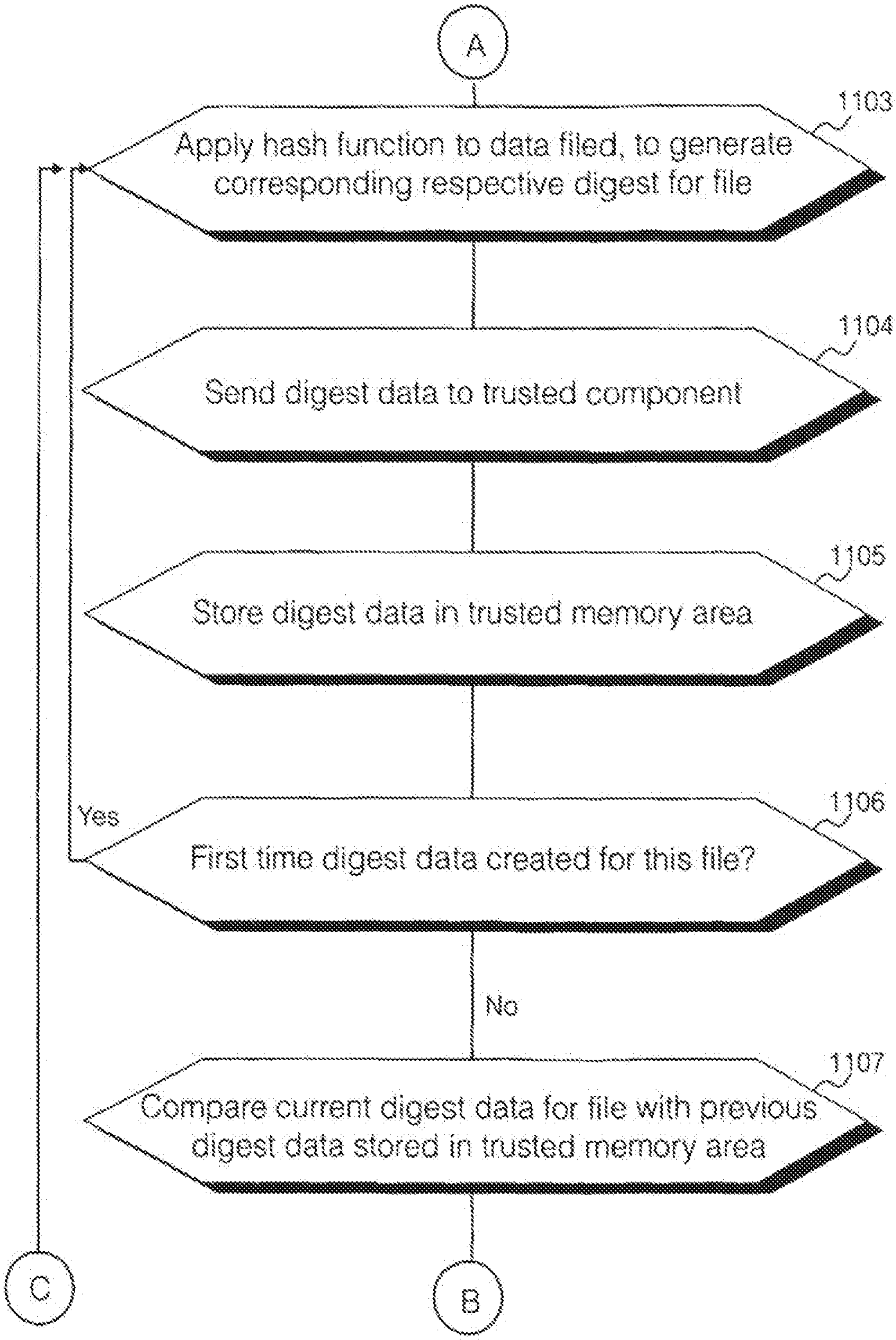

A
1103
Apply hash function to data filed, to generate corresponding respective digest for file
1104
Send digest data to trusted component
1105
Store digest data in trusted memory area
1106
Yes
First time digest data created for this file?
No
1107
Compare current digest data for file with previous digest data stored in trusted memory area
C
B

Fig. 11b

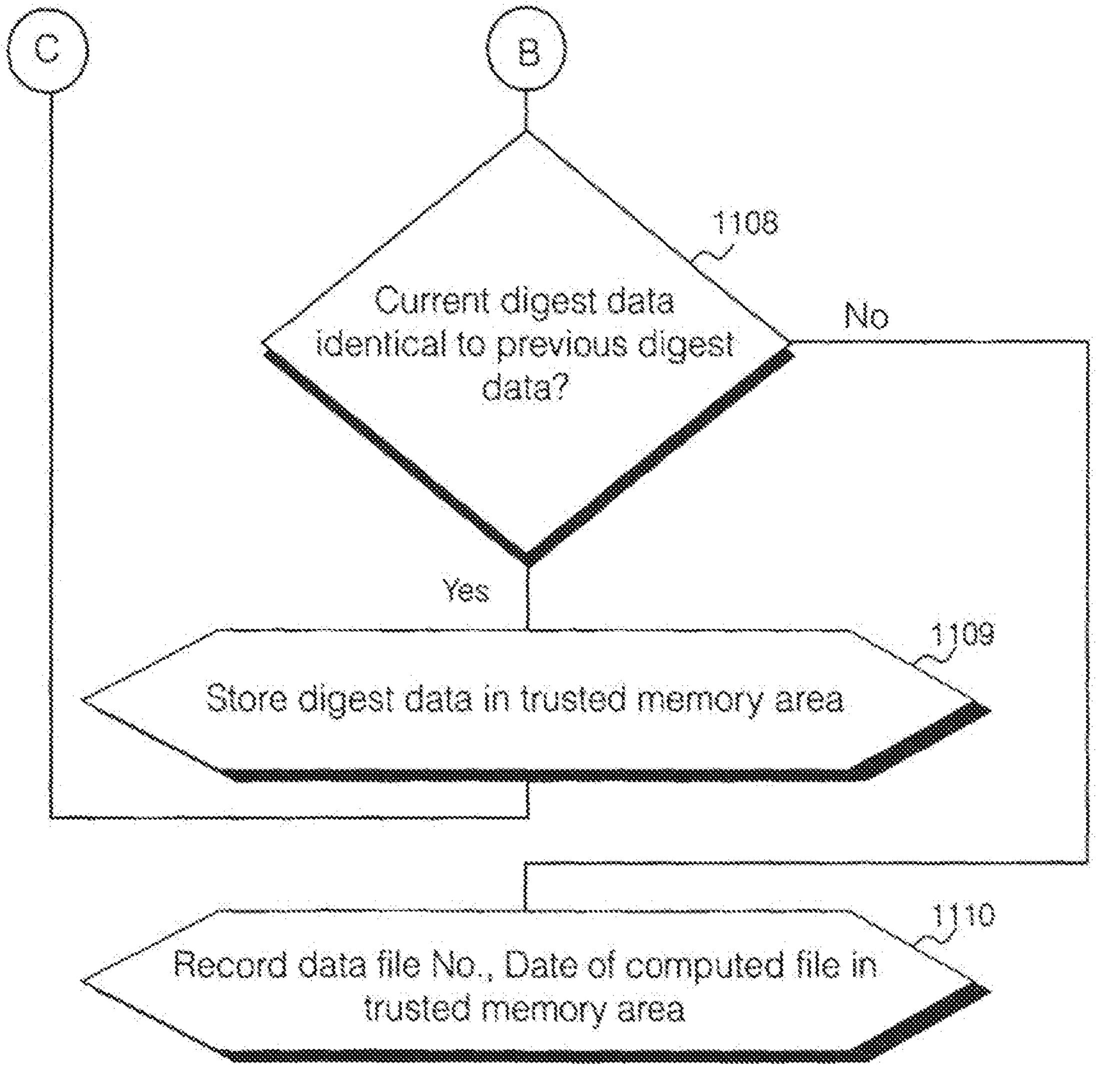

C
B
1108
Current digest data identical to previous digest data?
No
Yes
1109
Store digest data in trusted memory area
1110
Record data file No., Date of computed file in trusted memory area

Fig. 11c

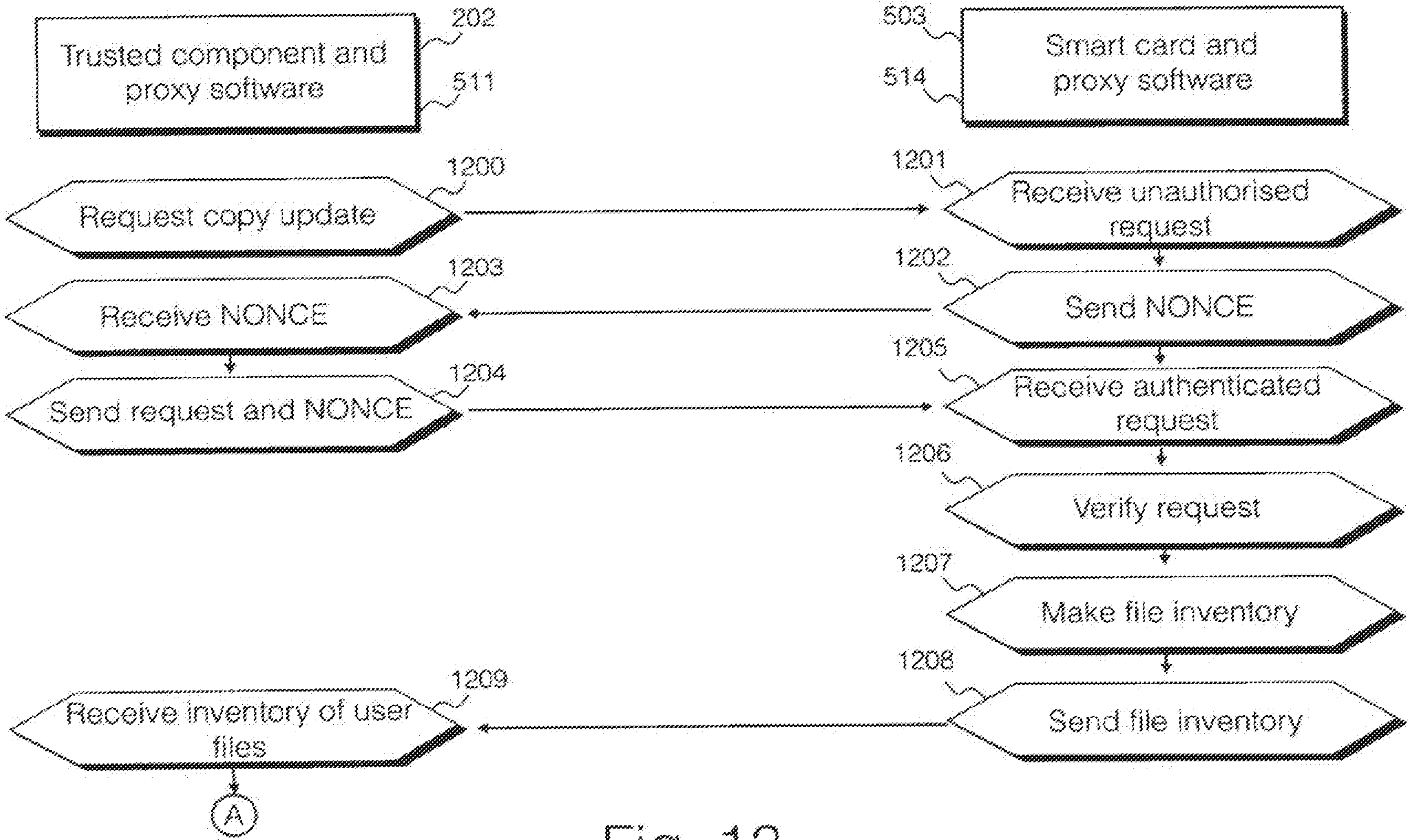

Trusted component and proxy software
202
511
Smart card and proxy software
503
514
Request copy update
1200
Receive unauthorised request
1201
Send NONCE
1202
Receive NONCE
1203
Send request and NONCE
1204
Receive authenticated request
1205
Verify request
1206
Make file inventory
1207
Send file inventory
1208
Receive inventory of user files
1209
A

Fig. 12

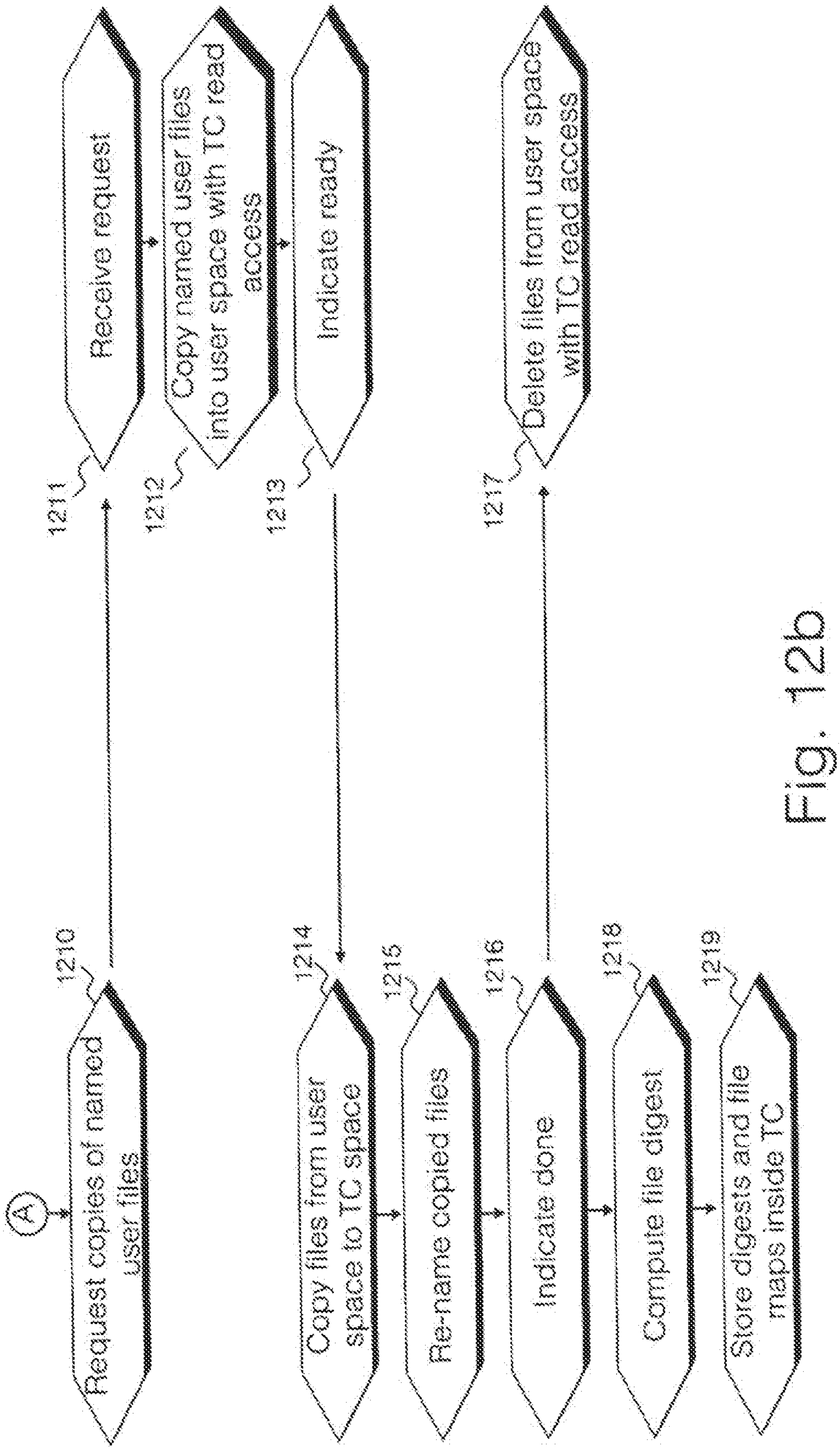

A
1210
Request copies of named user files
1211
Receive request
1212
Copy named user files into user space with TC read access
1213
Indicate ready
1214
Copy files from user space to TC space
1215
Re-name copied files
1216
Indicate done
1217
Delete files from user space with TC read access
1218
Compute file digest
1219
Store digests and file maps inside TC

Fig. 12b

European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 99 30 4166

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
X	"SYSTEM FOR DETECTING UNDESIRED ALTERATION OF SOFTWARE" IBM TECHNICAL DISCLOSURE BULLETIN, vol. 32, no. 11, April 1990 (1990-04), pages 48-50, XP000097602 armonk, usa * the whole document * -----	1-5,7	G06F11/00 G06F1/00 G06F12/14
A	V. BONTCHEV: "Possible Virus Attacks Against Integrity Programs And How To Prevent Them" VIRUS BULLEETIN CONFERENCE, September 1992 (1992-09), pages 131-141, XP000613974 Oxon, England * the whole document * -----	1-27	
A	US 5 421 006 A (JABLON ET AL.) 30 May 1995 (1995-05-30) * the whole document * -----	1-27	
A	US 5 572 590 A (CHESS) 5 November 1996 (1996-11-05) * abstract * -----	1-27	TECHNICAL FIELDS SEARCHED (Int.Cl.7) G06F
A	US 5 619 571 A (SANDSTROM ET AL.) 8 April 1997 (1997-04-08) * the whole document * -----	1-27	
A	US 5 359 659 A (ROSENTHAL) 25 October 1994 (1994-10-25) * the whole document * -----	1-27	
A	WO 93 25024 A (CYBERLOCK DATA INTELLINGENCE, INC) 9 December 1993 (1993-12-09) * the whole document * ----- -/--	1-27	

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	9 December 1999	Absalom, R

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 99 30 4166

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
A	Y. RADAI: "Checksumming Techniques for Anti-Viral Purposes" VIRUS BULLETIN CONFERENCE, September 1991 (1991-09), pages 39-68, XP000700179 Oxon, England		
A	EP 0 849 657 A (NCR INTERNATIONAL, INC.) 24 June 1998 (1998-06-24) * the whole document *	1-27	
A	US 5 844 986 A (DAVIS) 1 December 1998 (1998-12-01) * the whole document *	1-27	
A	EP 0 825 511 A (HEWLETT-PACKARD COMPANY) 25 February 1998 (1998-02-25)		
			TECHNICAL FIELDS SEARCHED (Int.Cl.7)

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	9 December 1999	Absalom, R

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 99 30 4166

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report.
The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

09-12-1999

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5421006 A	30-05-1995	NONE	
US 5572590 A	05-11-1996	NONE	
US 5619571 A	08-04-1997	NONE	
US 5359659 A	25-10-1994	NONE	
WO 9325024 A	09-12-1993	NONE	
EP 849657 A	24-06-1998	JP 10282884 A	23-10-1998
US 5844986 A	01-12-1998	AU 4146197 A EP 0932953 A WO 9815082 A	24-04-1998 04-08-1999 09-04-1998
EP 825511 A	25-02-1998	US 5841869 A JP 10154130 A	24-11-1998 09-06-1998

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82